UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/025,774	12/26/2001	Mauricio Lopez	BS01-321	5878

45695 7590 02/11/2005

WITHERS & KEYS FOR BELL SOUTH
P. O. BOX 71355
MARIETTA, GA 30007-1355

EXAMINER	
CHOW, CHIH CHING	
ART UNIT	**PAPER NUMBER**
2122	

DATE MAILED: 02/11/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/025,774	LOPEZ ET AL.	
	Examiner	Art Unit	
	Chih-Ching Chow	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 January 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 January 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *05/29/02*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to amendment dated January 10, 2005.

2. Per Applicants' request, claims 1, 5, 9, 13, 17 and 19 have been amended. Claims 1-21 remain pending.

Response to Amendment

3. Applicants' amendment dated 01/10/2005, responding to the 10/05/2004 Office action provided in the objection of drawings. The examiner has reviewed the updated specification, the specification has been amended to refer to 601 in order to match the Figure 6 drawing.

4. The set of formal drawings filed concurrently with the above-mentioned amendment is accepted by the Examiner.

5. Applicants' amendment dated 01/10/2005, responding to the 10/05/2004 Office action provided in the rejection of claims 5, 13 and 19 for use of the JAVA trademark incorrectly. The examiner has reviewed the updated claims respectfully.

6. The rejection to the claims 5, 13 and 19 are hereby withdrawn in view of Applicants' amendment.

Response to Arguments

7. Applicant's arguments for Claims 1-21 have been fully considered respectfully by the examiner but they are not persuasive. See MPEP 7.38 Arguments are Moot because of new ground(s) of rejection Applicant's arguments with respect to claims 1, 9, and 17 have been considered but are moot in view of the new ground(s) of rejection. The amended parts, 'acquiring program data that defines the underlying program structure including one or more object language

components of the application, 'displaying the program data including the one or more object language components to a maintenance person', and 'executing the command to cause the program data of the executing application to be modified without suspending or terminating the executing application", were never mentioned in the original claims, the Applicants have narrowed the limitation of the claims, therefore a new prior art is referenced in this office action.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless -
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-4, 9-12, and 17 are rejected under 35 U.S.C. 102(b) as being anticipated by "A Probe-Based Monitoring Scheme for an Object-Oriented, Distributed Operating System", OOPSLA, 09/1986, by Partha Dasgupta, (hereinafter, "Dasgupta").

CLAIM	Dasgupta
1. A method for modifying a computer application in substantially real-time without suspending or terminating the application, said method comprising the steps of: (a) connecting to an application executing on an application server, the application server having a computer memory; (b) acquiring program data that defines	Dasgupta teaches an example in using 'PROBE' for an application program. 'Probing' is a well-known computer software diagnostic technique to the people of ordinary skill in the art. Probe allows user to display and modify a designated program object while executing an application program (in real-time); the displaying and modifying is done at an independent window, which

the underlying program structure including one or more object language components of the application;

(c) displaying the program data including the one or more object language components to a maintenance person;

(d) accepting a command from the maintenance person; and

(e) executing the command to cause the program data of the executing application to be modified without suspending or terminating the executing application.

is an **user interface**, functioning as an **interactive console** for the user (*maintenance person*). Probing can display any designated variable contents, accepting arguments, it can also prompt user with a command line in order to modify a variable's content or any other commands. By modifying the variable's content, it can force the program to execute in an alternative path. See Dasgupta page 57, second paragraph, "The basic mechanism we propose, for monitoring, is the usage of probes. Probes are a powerful tool in many environments and has been proposed for deadlock detection, debugging, backup processing and so on" (*connecting to an application executing on an application server*), in Dasgupta page 57, first paragraph, "Give the ability to detect failing, flaky or failed components (software modules and hardware units) the system has the ability to reconfigure the healthy units, on the fly, to work around the faulty ones", also page 57, column 2, second paragraph, "The probe system also has some other payoffs like the easy implementation of **interactive** debugging support", and on page 62, column 1 last paragraph, "The monitor periodically **probes all the components in its list** (*acquiring program data that defines the underlying program structure, displaying the program data*). The status of these components are **stored in a fully**

replicated database. This database has the **same structure** and properties as the database used to locate Clouds objects,...it is **highly available**" -- probing works in real-time (without suspending or terminating the executing application), it **interacts** with the executing program and changes the program's behavior if the user has chose to do so (*accepting a command from the maintenance person, executing the command to cause the program behavior*); basically probing is able to give a snapshot of application program status (underlying program structure), it can manipulate the designated variable contents and thus change the program behavior, which is similar as described in the current application paragraph 0024, "these components are used to extract the fields and execute the methods of the object class. The fields of the object can be manipulated to create different behaviors in the object."

2. The method of claim 1, further comprising the step of modifying data that has been cached in the computer memory.

For the feature of claim 1 see claim 1 rejection. In Dasgupta, page 62, column 1, last paragraph, "The monitor periodically probes all the components in its list. The status of these components are **stored in a fully replicated database**. This database has the same structure and properties as the database used to locate Clouds objects,...it is **highly available**" to store the modifying data into cache memory

	for faster accessing is a design choice; it's an anticipated skill to those skilled in the art.
3. The method recited in claim 1, further comprising the step of modifying an order of the execution of a plurality of methods within the application.	For the feature of claim 1 see claim 1 rejection. On Dasgupta page 57, under Section 2, second paragraph, "The basic building blocks in Clouds are **objects**, actions and process (*methods*). Processes are carriers of the thread of control, on behalf of actions. The actions are atomic units of activity, consisting of a partial **order of invocations of operations defined in objects**".
4. The method recited in claim 1, further comprising the step of providing more detailed diagnostic messages in response to the command.	For the feature of claim 1 see claim 1 rejection. As mentioned it claim 1 rejection, PROBING allows user to set up commands at designated break points and to dump the program variable values, if the break points are set up appropriately, user can get more detailed diagnostic information when stepping through the break points.
9. A system for modifying an application in substantially real-time during execution without suspending or terminating the application, comprising: (a) an application server on which the application executes; (b) an object shell console that attaches to the application while it is running to obtain program data defining the underlying program structure of the	Probing is a 'system' which includes both software and hardware support. See Claim 1 rejection, the probing independent window, which attaches to the application while it is running to obtain designated program status, has the same function as the 'object shell console' in claim 9. For the rest of claim 9 feature see claim 1 rejection.

application including at least one object language component;
(b) a graphical user interface in the object shell console that is used to assist a maintenance person in modifying the program data of the application; and
(c) a command line for accepting a command to be executed, said command when executed will cause the execution of the application to be modified without suspending or terminating the application.

10. The system recited in claim 9, further comprising:
a vector for establishing an order of method execution in the application;
a command line for entering a new vector comprising a different order for executing the methods in the application; and
wherein entering the new vector in the command line establishes the different order of method execution.

For the feature of claim 9 see claim 9 rejection. See Claim 1 rejection, the 'probing components list' can be a list of commands, which is the same as a vector for establishing an order of method execution in the application.

11. The system recited in claim 9, further comprising a data cache in a memory of the application server, said data cache being modified by the command.

For the feature of claim 9 see claim 9 rejection. For the rest of the feature in claim 11, see claim 2 rejection.

12. The system recited in claim 9, further comprising:
a terse logging operation that provides terse diagnostic messages;
a detailed logging operation that

For the feature of claim 9 see claim 9 rejection. See claim 1 and claim 4 rejections.

provides detailed diagnostic messages; and
means for allowing a maintenance person to select detailed diagnostic messages if an error occurs without suspending or terminating the application.

17. A system for modifying an executing application in substantially real-time, comprising:
a computer on which the application is executing;
means for attaching to the executing application so that program data that defines the underlying program structure and that includes at least one object language component is extracted from the executing application;
a display device for displaying the program data to a maintenance person;
means for accepting a command from the maintenance person; and
means for invoking the command to thereby cause the program data of the application to be modified in accordance with the command without suspending or terminating the application.

Same as claim 1 rejection.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been

obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 5, 13, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over "A Probe-Based Monitoring Scheme for an Object-Oriented, Distributed Operating System", by Partha Dasgupta (hereinafter, "Dasgupta"), in view of 'Introduction to java Remote Method Invocation (RMI)', by Chris Matthews (hereinafter, "Matthews").

CLAIM	Dasgupta / Matthews
5. The method recited in claim 1, further comprising the step of connection to the application using Java RMI.	For the feature of claim 1 see claim 1 rejection. Dasgupta teaches object-oriented, Distributed Operating System; Dasgupta mentioned in page 58, 4th paragraph 'object is an instance of an abstract data type' (*class*), 'set of routines that can access' (*method*) - these are all Java concepts, however, Java does not exist yet in 1986. Dasgupta teaches all aspect in claim 5 but does not mention the 'connection to the application using Java RMI' specifically. However Matthews teaches this feature in a prior art. In Matthews, page 1, 4th paragraph, "**Java RMI** is shipped with the Java JDK 1.1 and higher. It is a true **distributed computing application** interface for Java". It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Dasgupta's disclosure of using probes by invoking a remote procedure taught by Matthews for the purpose of accessing

	remote objects over the network (see Matthews page 2, second paragraph).
13. The system recited in claim 9, wherein the object shell console attaches to the application using Java RMI.	For the feature of claim 9 see claim 9 rejection. For the rest of the feature of claim 9, see claim 5 rejection.
19. The system recited in claim 17, wherein the attaching means further comprises means for attaching to the application using Java RMI.	For the feature of claim 17 see claim 17 rejection. For the rest of the feature of claim 19, see claim 5 rejection.

12. Claims 6, 14, 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over "A Probe-Based Monitoring Scheme for an Object-Oriented, Distributed Operating System", by Partha Dasgupta (hereinafter, "Dasgupta"), further in view of 'JAVA in A Nutshell' by David Flanagan (hereinafter "Flanagan").

CLAIM	Dasgupta / Flanagan
6. The method recited in claim 1, further comprising the step of modifying an application written in an interpreter programming language.	For the feature of claim 1 see claim 1 rejection. Dasgupta teaches all aspects of claim 1 but does not mention the 'interpreter programming language' specifically. However, Flanagan teaches that feature in his book. Java is an **interpreter programming language**, see Oreilly, page 14, last paragraph, "To invoke a Java program, you run the Java **interpreter**, java," and page 247, "The **Java Inerpreter**" section. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement

	Dasgupta disclosure of the method using PROBing for Java language further taught by Flanagan for the purpose of invoking a Java program.
14. The system recited in claim 9, wherein the application is written in an interpreter programming language.	For the feature of claim 9 see claim 9 rejection. For the rest of the claim 14 features, same as claim 6 rejection.
18. The system recited in claim 17, wherein the application is written in an interpreter programming language.	For the feature of claim 17 see claim 17 rejection. For the rest of the feature of claim 18, see claim 6 rejection.

13. Claims 7-8, 15-16, 20-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over "A Probe-Based Monitoring Scheme for an Object-Oriented, Distributed Operating System", by Partha Dasgupta (hereinafter, "Dasgupta"), in view of 'Java Remote Method Invocation' 06/98, by Gopalan Suresh Raj (hereinafter "Raj"), and further in view of U.S. 6,681,389 by Norbert Engel et al. (hereinafter "Engel").

CLAIM

7. The method recited in claim 1, further comprising the steps of:

(a) accepting a selection of a method from the program data; and

(b) invoking the method from the command line with at least one new argument.

Dasgupta / Engel / Raj

For the feature of claim 1 see claim 1 rejection. Dasgupta teaches all aspects of claim 1 but does not mention the 'accepting a selection of method' and 'invoking method' (method is a Java term, Java does not exist yet in 1986) specifically. However, Engel teaches the 'accepting a selection' feature and Raj teaches 'invoking method' feature in an analogous art. In Engel, column 4, lines 61-65, "The user can **select** (there must be a list of software **prompted** for the user to do the selection) which application component to upgrade on one or more online machines/servers in the cluster. This provides the flexibility of updating a subset of application software rather than all application software on all machines/servers in a cluster." In Raj, page 1, 2nd paragraph, "In the RMI model, the server defines objects that the client can use remotely. The clients can now **invoke methods of this remote object** (***accepting*** *a selection of a method*) as if it were a local object running in the same virtual machine as the client. RMI hides the underlying mechanism of transporting method **arguments** and return values across the network." Further, last paragraph, "which in turn **calls the appropriate method** on the server object. In other words, the stub acts as a proxy to the skeleton and the skeleton is a proxy to the actual remote

	method." It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Dasgupta disclosure of the method using Probing by the feature of entering argument and invoking methods further taught by Engel and Raj for the purpose of passing parameters during method calls between machines (see Raj, page 2, 2nd paragraph).
8. The method recited in claim 1, further comprising the steps of: (a) accepting a selection of a method and the program data; (b) prompting the maintenance person for at least one new argument value; and (c) invoking the method with the at least one argument.	For the feature of claim 1 see claim 1 rejection. For the rest of the features in claim 8, see claim 7 rejection.
15. The system recited in claim 9, wherein the program data comprises at least one method related that is executed in the application, further comprising: means for accepting a selection of a method from the program data; and means for invoking the method from the command line with at least one new argument.	For the feature of claim 9 see claim 9 rejection. For the rest of the features in claim 15, see claim 7 rejection.
16. The system recited in claim 9, wherein the program data comprises at least one method related that is executed in the application, further	For the feature of claim 9 see claim 9 rejection. For the rest of the features in claim 16, see claim 1 and claim 7 rejections.

comprising:
means for accepting a selection of a method from the program data;
a prompt to prompt the maintenance person for at least one new argument value; and
means for invoking the method with the at least one new argument value.

20. The system recited in claim 17, wherein the program data includes one or more methods that are executing in the application, further comprising:
means for accepting a selection of a method from the display of the program data; and
means for invoking the method from the command line with at least one new argument.

For the feature of claim 17 see claim 17 rejection. For the rest of the features in claim 20, see claim 1 and claim 7 rejections.

21. The system recited in claim 17, wherein the program data includes one or more methods that are executing in the application, further comprising:
means for accepting a selection of a method from the display of the program data;
a prompt to prompt the maintenance person for at least one argument value; and
means for invoking the method with the at least one argument.

For the feature of claim 17 see claim 17 rejection. For the rest of the features in claim 21, see claim 7 rejection.

Conclusion

14. The following summarizes the status of the claims:

35 USC § 102 rejection : Claims 1-4, 9-12, and 17

35 USC § 103 rejection : Claims 5-8, 13-16, and 18-21.

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Miyamoto, U.S. Patent No. 6,662,363 discloses a method for generating a software installation system, which includes a user interface to allow user to make a selection of the software.

Greg Burns, "Cross-Debugging Real-Time Ada® Programs", discloses using Probe for debugging Ada® programs and functions and features for using Probe.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be

calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Even a final action is submitted, however the Examiner will re-evaluate the following claims, if the applicants can be more specific in the following claims:

Claim 1, 'displaying the program data including the one or more object language components to a maintenance person', and 'accepting a command from the maintenance person'. What are the possible commands that a maintenance person can enter based on the display? It is not clear what is in the 'object language components'? In spec paragraph 0024, the object language components only specified 'Class, Method and Field'.

Claim 9, what are the 'object language components'? Are they other object code or other external procedures? Also after the new object language component is obtained from the object shell console, how can the code be 're-executed', in paragraph 0025, "After the method is re-executed, the application will use the new value created by the method", the Applicant do not specify in the claim HOW can this piece of new code be 're-executed'? Does it get re-invoked via an internal timer pop? What is the command related to the 'object language component'?

Claim 17, similar as claim 9, 'program data that defines the underlying program structure and that includes at least one object language component is extracted from the executing application' - sounds like the program data is dumps of program variables? What is the program data related to a command? And how can the command be invoked and thus modify the program's execution?

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 7:00am - 3:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow

Examiner

Art Unit 2122

CC

ANTONY NGUYEN-BA
PRIMARY EXAMINER